High-fidelity aerodynamic shape optimization using efficient orthogonal modal design variables with a constrained global optimizer

D. J. Poole[1], C. B. Allen[2], T. C. S. Rendall[3]

Department of Aerospace Engineering, Queens Building, University Walk, University of Bristol, Bristol, BS8 1TR, U.K.

Abstract

Aerodynamic shape optimization of aerofoils using efficient orthogonal design variables is considered using a global search algorithm. A novel approach is presented for deriving shape design variables, using a proper orthogonal decomposition of a set of training aerofoils to obtain an optimally efficient set of aerofoil deformation modes that represent typical design parameters such as thickness and camber. A major advantage of this extraction method is the production of orthogonal design variables, and this is particularly important in aerodynamic shape optimization. These design parameters have previously been tested on geometric shape recovery problems and been shown to be efficient at covering a large portion of the design space, hence the work is extended here to consider their use in aerodynamic shape optimization. A global search algorithm with an efficient constraint handling method has been developed and used here to optimize a suite of inviscid **and viscous** compressible aerofoil test cases using varying numbers of modal parameters. Often, an artefact of inviscid optimizations is an oscillatory pressure distribution, so to alleviate this drag minimization with a modulus of curvature penalty is also considered for the inviscid optimizations, where the penalty is used to force smoother pressure distributions; **this is not necessary in the viscous optimizations.** Results indicate that often fewer than 10 design parameters are required to obtain shock free solutions even from highly-loaded aerofoils with significant shocks.

Keywords: Aerodynamic shape optimization, Singular value decomposition, Shape parameterization, Global optimization

1. Introduction and Background

Aerodynamic shape optimization (ASO) is the process often used to optimize a given aerodynamic shape within a computational environment to improve on a design requirement. Numerical simulation methods to model fluid flows are used routinely in industrial design, and increasing computer power has resulted in their integration into the optimization process to produce the ASO framework. The aerodynamic model is used to evaluate some metric against which to optimize, which in the case of ASO is an aerodynamic quantity, most commonly drag, subject to a set of constraints which are usually aerodynamic or geometric. Along with the fluid flow model, the ASO framework requires a surface parameterization scheme, mathematically describing the aerodynamic shape being optimized by a series of design variables. Changes in the design variables, which are made by a numerical optimization algorithm, result in changes in the aerodynamic surface. Numerous advanced optimizations using compressible computational fluid dynamics (CFD) as the aerodynamic model have previously been performed for aerofoil sections [1, 2], full aircraft [3, 4, 5], aeroelastic aircraft [6], and rotor blades [7, 8, 9, 10]. The authors have also presented work in this area, having developed a modularised, generic optimization tool, that is flow-solver and mesh type independent, and applicable to any aerodynamic problem [11, 12].

The fidelity of results obtained by the optimization process is dependent on the fidelity and quality of each of the three individual components of the ASO process. To facilitate optimum compatibility between these components, each is often designed in a modular manner such that, for example, the aerodynamic model is independent of the parameterization scheme used. A high fidelity numerical aerodynamic model with good capturing of the true physics is important in producing optimum aerodynamic designs, particularly at transonic conditions. The aerodynamic model also defines the parameter space of the problem, which is the definition of the aerodynamic outputs based on flow field inputs such as Mach number and angle of attack.

The quality of the optimization result obtained is driven primarily by the quality and type of

Email addresses: dp8470@bristol.ac.uk. (D. J. Poole), c.b.allen@bristol.ac.uk. (C. B. Allen), thomas.rendall@bristol.ac.uk. (T. C. S. Rendall)

[1]Graduate Student.

[2]Professor of Computational Aerodynamics. Corresponding author.

[3]Lecturer.

numerical optimization algorithm used in the ASO framework. The two primary types of optimization algorithms are local methods and global methods. The local methods are usually built around the gradient-based approach, which uses the local gradient of the design space as a basis around which to construct a search direction. The optimization algorithm therefore traces a movement path through the design space until the gradient values become very small where the result has converged. These approaches are the most common methods used in the ASO framework ([13, 14, 5] for example), driven primarily by the low cost associated with them compared to global methods [15]. Global optimization algorithms, on the other hand, tend to be based around a swarm intelligence approach, where candidate solutions scattered throughout the design space cooperate together to locate the global optimum solution. Each candidate solution, of which there are typically around the order of 100, requires an objective function evaluation, which in ASO is a flow solution, and therefore are often considerably more expensive than the local algorithms. Furthermore, handling of constraints using global methods can be difficult and is often done on an *ad hoc* basis. The primary advantage, however, is that global algorithms are much less prone to converging in locally optimum solutions that are not necessarily close to the global optimum. Due to the high cost associated with such algorithms, and their issues in handling constraints (which is an important consideration for ASO), their use in ASO is more restricted than local methods but is becoming more common [16, 17, 18].

The aerodynamic model defines the parameter space of the problem, however, the problem design space, which the optimization algorithm interrogates, is constructed by the definition of a surface parameterization scheme. The ability of the optimizer to fully interrogate the true design space (which contains every possible design) is driven by the ability for the degrees of freedom of the parameterization scheme to represent any shape within the design space, and so this is a critical aspect of any optimization scheme. Furthermore, the use of a low number of design variables is highly advantageous, particularly if global optimization algorithms are used where the so-called 'curse of dimensionality' means that performance of global search algorithms deteriorates with increasing dimensions.

An important aspect of any parameterization scheme is orthogonality of the design variables. Orthogonal design variables means that a shape is represented by a unique set of inputs, often

leading to a design space that is more efficient, meaning it can be represented with fewer design variables [19]. It also tends to simplify the design space against non-orthogonal design variables and leads to greater coverage of the design space, i.e. the design variables can represent a greater number of aerodynamic shapes; the design space of N design variables is always contained within the design space of $N + n$ design variables.

The work presented in this paper considers the optimization of aerofoils using a novel method of deriving design variables. Proper orthogonal decomposition (POD) is used on a training library of aerofoils, the selection of which can be considered a type of dimensional filtering, to mathematically extract optimum aerofoil shape modes, and these shape modes have the major advantage of being orthogonal. **The method, which has recently been presented by the authors [20], was shown to be highly effective at representing the aerofoil design space often requiring less than a dozen variables to recovery aerofoil shapes. Furthermore, Masters *et al.* [21] demonstrated that for aerofoil design space representation, deriving aerofoil modes by SVD produced the most efficient and compact set of design variables, outperforming many other commonly used aerofoil parameterization schemes.** The aim of the work presented here is to analyse the effectiveness of these novel orthogonal design variables when applied to aerodynamic optimization. Using POD means that advanced global search algorithms can be introduced into the ASO process; this has future implications on allowing investigations into aerodynamic multimodality. Hence, in this paper, an advanced constrained global search algorithm with an effective constraint handling framework is also employed to allow full design space exploration and exploitation. The use of a global search algorithm also means that any issue of multimodality in the design space can be effectively handled, thereby ensuring a true test of the design parameters.

2. Aerofoil Parameterization and Deformation Approaches

A surface parameterization scheme defines a design space by a number of design variables. A separate problem to this, though often considered alongside, is the deformation of the subsequent surface during the optimization process, which is required to allow deformation of a body-fitted CFD mesh. The effectiveness of a parameterization method is i) being flexible and robust enough to cover the design space, and ii) efficient enough to represent a given shape with as few design

variables as possible. Methods are classified as either constructive, deformative or unified. These are outlined below but more in-depth reviews have been presented by Samareh [22, 23], Nadarajah *et al.* [19, 24] and Masters *et al.* [25].

Constructive methods are those that consider the definition of the surface and the deformation of the surface separately. Examples of these methods are CST [26], PARSEC [27], PDEs [28] and splines [29]. Other approaches that combine various parameterizations in a hybrid approach, such as that of Zhu and Qin [30] can also be found. Due to the constructive nature of these approaches, perturbation of the base geometry through the optimization process therefore requires that the new surface be reconstructed which subsequently requires automatic mesh generation tools for production of a new surface and volume mesh. This extra difficulty can make it advantageous to consider approaches that manipulate an existing mesh.

An alternative to constructive are deformative methods which unify the geometry creation and perturbation. This tends to make them simpler to integrate with mesh deformation tools and allows the use of previously generated meshes – a considerably cheaper alternative to regeneration – though the mesh deformation is a separate algorithm. Analytic [1, 31] and discrete [32] methods are examples of deformative approaches.

A further refinement of combining geometry creation and perturbation is the integration with a mesh deformation algorithm, and these types of methods are unified. Methods of this type typically have some interpolation that describes a link between the surface and volume, often via a set of control points that are independent of both, such that deformation of the control points results in deformation of the surface and CFD mesh. These are the most common approaches found in ASO, and the methods included in this unified category are free-form deformation [33], domain elements[11] and direct manipulation [34].

A novel method, recently developed by the authors, is to consider deriving aerofoil design variables using a matrix decomposition approach [20]. The approach utilises singular value decomposition (SVD) in a manner that analyses an initial library of aerofoils and decomposes that library into a set of optimum, reduced design variables that are geometrically orthogonal to each another. Masters *et al.* [25] has shown the method to be able to represent the boundary shape of a wide range of aerofoils using a small subset of design variables. It was also shown that for this

inverse geometric design problem, that the modal design variables outperformed other commonly used aerofoil parameterization methods in terms of a minimum number of design parameters required to represent the boundary shape. This emphasises that having orthogonal design variables, which the SVD method provides, results in a more effective and efficient coverage of the aerofoil design space.

The SVD method can be formulated in either a constructive or deformative manner, and the deformative formulation is used in this work. It can also be linked to a unified approach to allow the design variables to be applied to the unified method, which is also the method used in this work to allow application of the parameters to ASO. Previous work has only considered the ability for the method to represent a wide-range of aerofoil shapes [20, 25], however, this work extends the method from just a geometric test to benchmarking against a suite of aerodynamic optimization problems.

3. Parameterization Scheme

The orthogonal design variables used here are based on aerofoil surface deformations and are linked to a unified parameterization scheme which uses control points to deform the CFD mesh, which is also called the domain element approach. The method for deriving the mathematical design variables is outlined first, followed by the control point-based approach for deforming the CFD mesh.

3.1. Aerofoil Deformations by Singular Value Decomposition

The derivation of aerofoil perturbation modes come from a proper orthogonal decomposition (POD), via singular value decomposition (SVD) of a training library of aerofoils. **The SVD decomposes a matrix into three constituent matrices and orders the dominant features of the input matrix. Hence, the SVD can be used to project a reduced order basis approximation and produce a low-rank approximation to the original input matrix. As shown by Eckart and Young [35], given a low rank approximation found through SVD, $\mathbf{A}^{(k)}$, of a full rank input matrix, $\mathbf{A}$, the following is true:**

$$\|\mathbf{A} - \mathbf{A}^{(k)}\|_F \leq \|\mathbf{A} - \mathbf{B}\|_F \tag{1}$$

where B is any matrix of rank k and $\|\cdot\|_F$ is the Frobenius norm. Hence, the error between the low rank approximation (found from SVD) and the full rank matrix will always be at least as good as the error between any other rank k matrix and the full rank matrix. In this sense, the SVD produces an optimal low order projection of the higher dimensional space into the lower dimensional one, which is an important result.

In this work, the SVD is used to decompose a training library of aerofoils to produce optimal aerofoil modes. The modes, which form aerofoil design variables used in this work for ASO, are guaranteed to be orthogonal, meaning a given aerofoil shape is described uniquely by a given set of input parameters. This alleviates some multimodality that can be introduced numerically by the given parameterization scheme, and expands design space coverage [19]. An alternative to deriving design variables by a direct decomposition approach is to manipulate already existing ones by Gram-Schmidt orthogonalisation to force orthogonality [36, 37], however, it is ideal to use the SVD method since the resulting low dimensional approximation (modal parameters) to a high dimensional design space (full training library) will be optimal, as shown in equation 1 i.e. no other set of aerofoil deformation parameters will be able to represent the original aerofoil library better than the SVD modes. Initial studies of the feasibility of using the SVD method to derive design variables have previously been demonstrated by Toal *et al.* [38] and Ghoman *et al.* [39], however, the work here develops more fully their use for performing aerodynamic shape optimization.

The SVD method requires a training library of m aerofoils to be collated from which the aerofoil deformation modes are extracted [20]. This library can be chosen based on some aerodynamic metric, and this can be considered a form of dimensional filtering. However, the suitable selection of the training library is of paramount importance in producing effective design variables. It has been shown that selecting a training library that is made primarily of transonic aerofoils results in more efficient modes for performing transonic design [20]. In general, it is best to select a training library that represents aerofoils with performance designed for the type of flow being considered.

The SVD method requires each aerofoil surface be parameterized by N surface points, where

the i-th surface point has a position in the space (x_i, z_i). To ensure consistency of the surface description of the training data all aerofoils are parameterized with the same parametric distribution, followed by each aerofoil having a rigid body translation, scaling and then rotation applied to it to map the geometry into a consistent form where the leading edge is located at the origin and the trailing edge at unit chord along the horizontal axis. A matrix is built from which SVD is performed, by evaluating the vector difference of the i-th surface point between all aerofoils, producing $m_{def} = m(m-1)/2$ aerofoil deformations. The x and z deformations are stacked into a single vector of length $2N$, for each aerofoil deformation, so a matrix is built of the aerofoil deformations which has $2N$ rows and m_{def} columns:

$$\Delta\mathbf{X} = \begin{pmatrix} \Delta x_{1,1} & \cdots & \Delta x_{1,m_{def}} \\ \vdots & \ddots & \vdots \\ \Delta x_{N,1} & \cdots & \Delta x_{N,m_{def}} \\ \Delta z_{1,1} & \cdots & \Delta z_{1,m_{def}} \\ \vdots & \ddots & \vdots \\ \Delta z_{N,1} & \cdots & \Delta z_{N,m_{def}} \end{pmatrix} \tag{2}$$

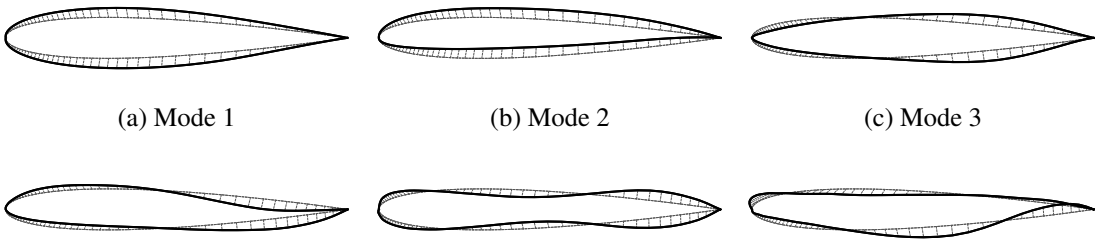
Performing an SVD decomposes the matrix into three constituent matrices:

$$\Delta\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3}$$

where $\mathbf{U}$ is a matrix of vectors, each of length $2N$. The structure is analogous to the decomposed matrix, so the columns of this matrix are the aerofoil mode shapes. $\mathbf{\Sigma}$ is a diagonal matrix of the singular values, arranged in descending order. These can be considered the 'relative energy' of the modes, and represent the 'importance' of the mode shapes in the original library. The total number of possible mode shapes is governed by the number of singular values, which is the minimum of the number of columns and rows of the decomposed matrix. A truncation of the $\mathbf{U}$ matrix, based on a certain total energy required, then gives the number of design variables used in the optimization. The training library is based on deformations and this is an important choice such that design variables that result from the decomposition are also deformations, ensuring they are independent of the class of the aerofoils that are used. This allows direct insertion into an aero-

dynamic shape optimization framework where deformation of the surface and mesh is important. If the constructive formulation is used, however, then the columns of the training matrix, $\Delta\mathbf{X}$, are absolute positions of the aerofoil surface points as opposed to deformations between surface points.

In this work, two different training libraries are considered based on the optimization being performed. The first is a generic, non-symmetric library containing 100 different aerofoils containing a variety of performances in the transonic regime. As outlined above, it is important to consider a training library containing aerofoils that have been designed to perform well in the flow regime that is being considered. The second library contains entirely symmetric aerofoils such that performing the SVD results in symmetric modes. The first six modes of the generic library are shown in figure 1, while the first six symmetric modes are shown in figure 2. It should be noted that all modes are scaled up for illustration purposes, and as such, sometimes crossover of the upper and lower surfaces is observed. This is unlikely to happen during the optimization process as the magnitude of the deformations tends to be much smaller than is shown here. If, however, the surfaces do cross, then the solution is an infeasible one and is handled as such by the optimization algorithm described below.



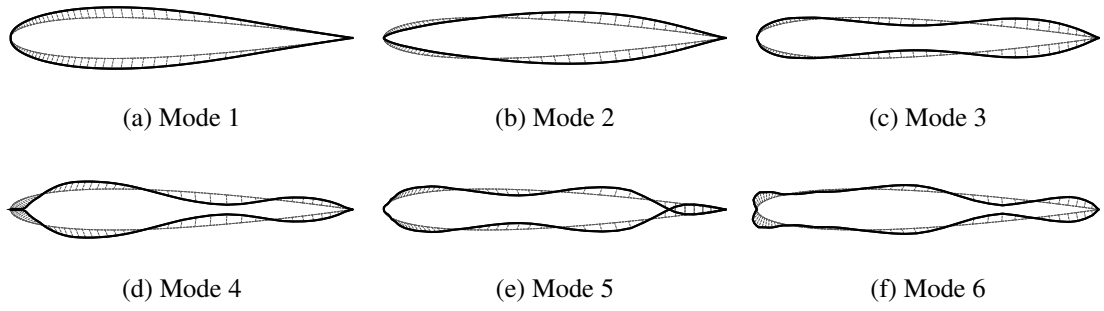
(a) Mode 1 (b) Mode 2 (c) Mode 3

(d) Mode 4 (e) Mode 5 (f) Mode 6

Figure 1: Generic non-symmetric aerofoil modes.

The raw surface data of the aerofoils is from an online database[4]. However, while the database in question contains a large archive of aerofoil sections, the data of the aerofoils is of mixed quality. For example, figure 3 shows the surface gradient of two example aero-

[4] http://m-selig.ae.illinois.edu/ads/coord_database.html



(a) Mode 1 (b) Mode 2 (c) Mode 3

(d) Mode 4 (e) Mode 5 (f) Mode 6

Figure 2: Symmetric aerofoil modes.

foils from the database demonstrating this. Smoothing can be performed on the database to obtain aerofoil surfaces of better quality (such as that done by Masters *et al.* [21]) but smoothing of the sections can lead to surfaces that are not exactly defined by the data points from the database. Therefore for this work, to ensure consistency with the online database, the definition of the aerofoil surfaces is forced through the raw data. To ensure a consistent parameterization of the aerofoil sections (needed for the SVD), a cubic b-spline is fit through the data points and a set of surface points is generated that have a consistent parametric distribution through the entire aerofoil library. The aerofoils are all normalized such that the leading edge is at the origin and the trailing edge is at unity along the horizontal coordinate. Only sharp trailing edge aerofoils are considered.

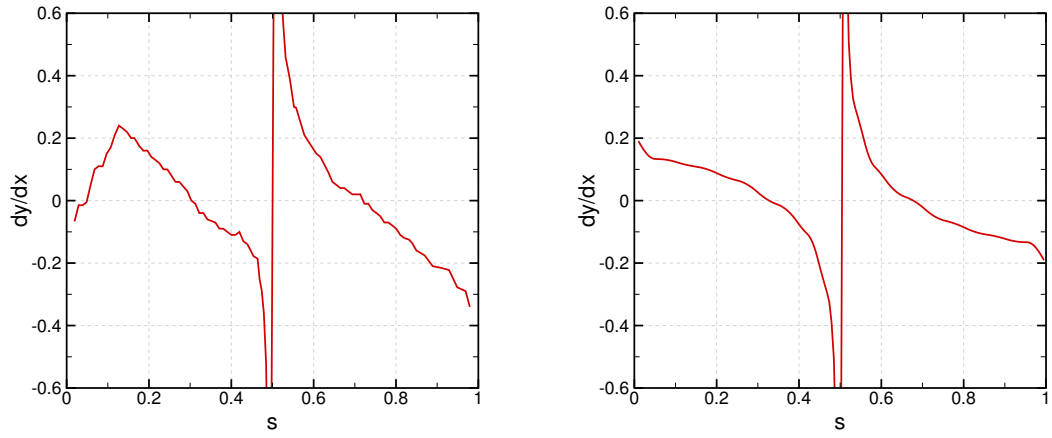


Figure 3: Raw surface gradients along normalised periphery of two aerofoils from online database.

Once the design variables have been extracted, and the total number of modes has been truncated then a new aerofoil is formed by a weighted combination of the modal parameters, as shown in equation 5. The weighting vector, $\boldsymbol{\beta}$, represents the magnitude of the modal deformations which are then the design variable values that the optimizer works with. The truncation of the total number of modes, which is often very large, down to a number which is useful for the optimization can either be done by a user specified number or based on the requirement for a total amount of energy to be preserved i.e. if 99.0% of the energy of the original library is required to be preserved then the first, say, six modes may cumulatively have 99.1% of the energy so six modes would be used. Alternatively, in this work, a number of modes is specified and those modes with the highest amount of energy are taken i.e. if six modes are used then these are the six modes with the highest energy. All modes are scaled so that their maximum deformation is the same.

$$\begin{pmatrix} x_{1,1}^{new} \\ \vdots \\ x_{N,1}^{new} \\ z_{1,1}^{new} \\ \vdots \\ z_{N,1}^{new} \end{pmatrix} = \begin{pmatrix} x_{1,1}^{old} \\ \vdots \\ x_{N,1}^{old} \\ z_{1,1}^{old} \\ \vdots \\ z_{N,1}^{old} \end{pmatrix} + \sum_{j=1}^{m_{modes}} \beta_j \begin{pmatrix} U_{1,j} \\ \vdots \\ U_{N,j} \\ U_{N+1,j} \\ \vdots \\ U_{2N,j} \end{pmatrix} \tag{4}$$

$$\mathbf{X}^{new} = \mathbf{X}^{old} + \sum_{j=1}^{m_{modes}} \beta_j \mathbf{U}_j \tag{5}$$

3.2. RBF Coupling of Point Sets for Aerofoil Deformation

The SVD method is used to derive a set of aerofoil design variables, however, these are surface deformations and must therefore be coupled to a control point-based approach to allow flexible deformation of the CFD mesh. The control point method links deformations of the CFD mesh to deformations of a small set of control points on or near the surface. At the centre of this technique is a multivariate interpolation using radial basis functions (RBFs), which provides a direct mapping between the control points, the surface geometry and the locations of grid points in the CFD volume mesh. The approach is meshless, so requires no connectivity and is applicable to any mesh type; control points and volume mesh points are simply treated as independent point

clouds. The system is only the size of the number of control points, and so is not related to the mesh size.

The general theory of RBFs is presented by Buhmann[40] and Wendland [41], and the basis of the method used here is described in detail by Rendall and Allen [42]. Let $f(\mathbf{x})$ be the original function to be modelled, and f_i be the scalar values at n discrete points $\mathbf{x}_i, i = 1, ..., n$, where $\mathbf{x}_i$ is the vector of inputs at the i-th sample point in d-dimensional space $\mathbf{x} = \{x^1, ..., x^d\}$. The set of data points $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is confined to a domain Ω in d-dimensional space. A RBF model is then a linear combination of basis functions, whose argument is the Euclidean distance between the point $\mathbf{x}$ at which the interpolation is made and the n points in the known data set. If ϕ is the chosen basis function and $\|\cdot\|$ is used to denote the Euclidean norm, then an interpolation model s has the form

$$s(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) \tag{6}$$

where $\alpha_i, i = 1, ..., n$ are model coefficients, and $p(\mathbf{x})$ is an optional polynomial. The polynomial term is excluded in the current work, since these will transfer unwanted deformations throughout the entire mesh. The coefficients are found by requiring exact recovery of the original data, $\mathbf{s}_\mathcal{X} = \mathbf{f}$, for all points in the training data set $\mathcal{X}$. Hence the model is an interpolant, and all original solution information is preserved.

Control points (sometimes named domain element points) are used here to decouple the shape parameters from the surface mesh, and provide a flexible framework through which to control the shape of a base geometry. Setting up a global RBF volume interpolation for n_c control points then requires a solution to a linear system, see [11] for more details, to ensure exact recovery of the control point deformations:

$$\Delta\mathbf{x}_c = \mathbf{M}\boldsymbol{\alpha}^x \tag{7}$$

These define exact recovery of the data, $\Delta\mathbf{x}_c$ (analogous definitions hold for y and z), where

$$\Delta \mathbf{x}_c = \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_{n_c} \end{pmatrix}, \quad \boldsymbol{\alpha}^x = \begin{pmatrix} \alpha_1^x \\ \alpha_2^x \\ \vdots \\ \alpha_{n_c}^x \end{pmatrix} \tag{8}$$

The linear system defined by equation 7 is then solved to find the unknown coefficients.

For surface and volume mesh deformation, it is sensible to use decaying basis functions, to give the interpolation a local character and ensure deformation is contained in a region near the moving body, and Wendland's C^2 function [41] is used here.

Hence, for the case considered here (which is in two dimensions) the global influence on any point in the aerodynamic mesh from the control points is determined by equation 6, which is applied as

$$\Delta x_a = \sum_{i=1}^{n_c} \alpha_i^x \phi(\|\mathbf{x}_a - \mathbf{x}_i\|) \tag{9}$$

$$\Delta z_a = \sum_{i=1}^{n_c} \alpha_i^z \phi(\|\mathbf{x}_a - \mathbf{x}_i\|) \tag{10}$$

where the subscript a refers to the aerodynamic mesh and n_c is the number of control points. The control point perturbations represent the design variables, which are decoupled from the surface and volume meshes.

3.3. Control Point Deformations

The method for deriving surface design parameters and the methods for perturbing the CFD mesh have been presented. The derived parameters are, however, surface deformations but for the aerodynamic optimization process, control point parameters are required. Previous work has involved placing control points away from the surface, to form off-surface domain elements, and this has proven very effective. Here, the control points to define the modal deformations are located on the surface of the aerofoil section. This ensures that there is direct coupling between the control point deformations and the surface deformations that derived them. To ensure orthogonality is preserved, the procedure of deriving the deformations by SVD is performed on the training

library parameterized by the number of control points and this is demonstrated in figure 4. **To ensure sufficient fidelity of representation of the modal deformations on the surface, a suitable number of control points must be used. Masters *et al.* [21] demonstrated that when using RBF control points on the surface for geometric shape recovery, that 20 to 25 control points provided a good fidelity of surface representation. While RBFs are not being used for surface recovery, but instead for mesh deformation, the general trend is still a useful one. Hence, 24 control points are used in this work.** Individual deformations of the control points are also possible as design variables (these are tested in this work), and this is demonstrated in figure 5.

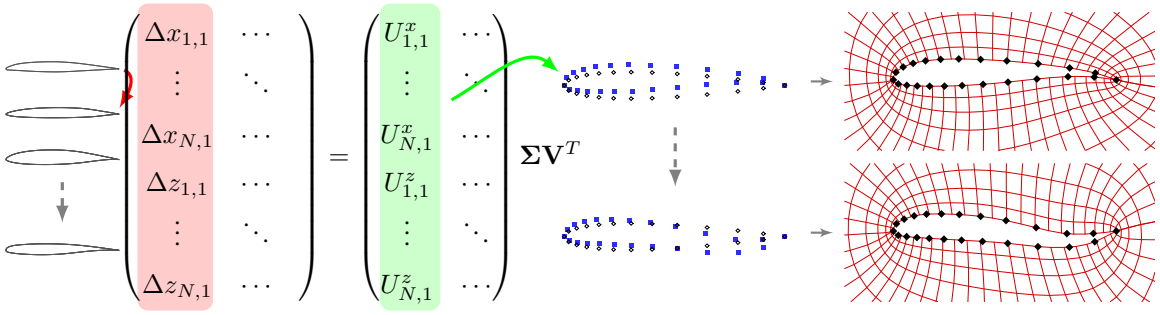


Figure 4: Process of obtaining control points deformations from SVD.

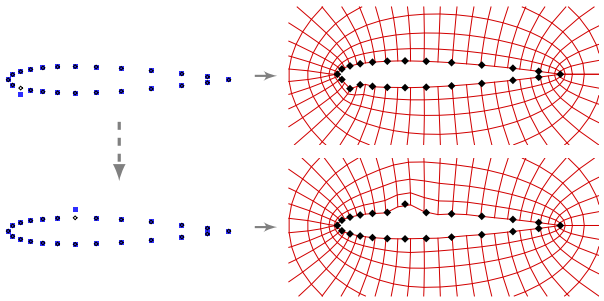


Figure 5: Illustration of individual control point deformations.

4. Optimization Algorithm

The aim of the work presented in this paper is to test the performance of the aerofoil design variables derived from SVD on aerodynamic optimization problems. Typically, the two main types of numerical optimization algorithm that are chosen for aerodynamic optimization are gradient-based and global search. Gradient-based methods, such as conjugate gradient and sequential quadratic programming (SQP) use the local gradient as a basis from which to construct a search direction. The algorithm starts at an initial solution and marches in a direction towards to minimum solution. Global search methods, however, use a number of agents with different starting positions within the search space. These agents then cooperate and move by various, often nature inspired mechanisms, towards the global optimum solution. Examples of gradient-based [43, 44, 45, 46] and global search [47, 16, 17] optimization algorithms within the ASO environment can be found.

The suitable selection of a gradient-based or a global search algorithm to perform aerodynamic optimization is highly dependent on the optimization case analysed, specifically the degree of modality present in the situation. Multimodal problems are characterised by multiple local optima, where one or more of those local optima is the globally optimum solution. This can be particularly problematic for gradient-based optimizers due to premature convergence in a local minimum that is not necessarily the global optimum. Agent-based methods can alleviate this issue somewhat. Within the context of aerodynamic shape optimization, the presence of a multimodal search space is highly dependent on the extent of the surface representation and the fidelity of the flow analysis tool. The issue of degree of multimodality in aerodynamic optimization problems is an unanswered question, with work presented showing that multimodality exists in a number of cases, but unimodal cases also exist [48, 49, 50, 15]. Chernukhin and Zingg [15] have considered this issue by testing a number of different optimization problems **with both gradient-based and global optimizers** and shown that for a b-spline parameterization of the surface, viscous, compressible drag minimization of the RAE2822 aerofoil has one global optimum **i.e. the gradient-based approach converged to the same answer irrespective of starting location, as did the global optimizers**. However, they also showed multiple local optima for other problems **using the same optimizers**.

For this work, to minimise the issue of multimodality, a global search algorithm is employed to perform aerodynamic optimizations. This ensures a true test of the new, orthogonal design parameters and therefore gives a fair comparison of their performance. Global search algorithms are more expensive compared to gradient-based approaches, however, they alleviate the issues of convergence in local minima that gradient-based methods are prone to. This is not to say that gradient-based methods are not a good choice for aerodynamic optimization, but to ensure greater robustness of the results, a global search algorithm has been chosen for this work.

An important consideration when using global optimization is the dimensionality of the problem. The so-called 'curse of dimensionality' states that the performance of global search algorithms deteriorates with increase dimensionality. This was outlined by van den Bergh and Engelbrecht [51] who used the analogy of random sampling to try to locate the optimal region (i.e. the approximate region of the globally optimal solution). The probability that the optimal region is found through random sampling is given by the volume of the optimal region divided by the volume of the whole design space. If it is assumed that the design space is a unit hypercube of d dimensions, and the optimal region is a hypercube with some length that is a fraction, η, of the length of the total design space, then the probability of randomly locating the optimal region is η^d. If, say, $\eta = 0.01$ (i.e. the optimal region is 1% of the total design space in each dimension) then for a five dimensional design space, the probability is 1×10^{-10}. While the performance of global search algorithms alleviates this problem considerably, the trend of decrease performance with increase dimensionality still holds true. It is therefore important, when using global search algorithms, to have a low dimensional design space. In this work, the orthogonal design parameters derived by SVD allow this.

The specific global search algorithm that is employed is the gravitational search algorithm (GSA) [52], which is an agent-based approach that is based on Newtonian gravitational laws and mechanics, where a set of agents in the search space have a fictitious mass depending on their fitness, and gravitational attractive forces transfer data throughout the swarm and move the agents towards other, 'heavier' (good fitness) agents. Constraints are handled by the separation-sub-swarm approach (3S) framework [53] which separates out infeasible and feasible particles into separate swarms. The infeasible particles minimize the constraint violation and the feasible

particles minimize the objective function. For completeness, the full 3S-GSA algorithm is outlined below, **though full details of the development and thorough testing of the constraint handling framework with benchmarking against other commonly used constraint handling frameworks (such as penalty functions) can be found in [53]**.

4.1. 3S-GSA Algorithm

The algorithm requires a system of N agents to be initialised within the bounds of the design variables, where the position of the n-th agent is denoted by:

$$\mathbf{x}_n = \{x_n^1, x_n^2, \ldots, x_n^d\}^T \tag{11}$$

where x_n^1 represents the position of the n-th particle in dimension 1. In the design problem considered here, this would be β_1 in equation 5 i.e. the magnitude of the first modal deformation. The velocity of the agents is also initialised to be a random value that is smaller than the bounds of the search space. For each agent, the objective function and the values of the constraints need to be calculated. The objective function associated with the n-th agent is:

$$\overline{f(\mathbf{x}_n)} = \begin{cases} f(\mathbf{x}_n) & \text{if } \mathbf{x_n} \text{ is feasible} \\ \sum_{j=1}^{G} \max\{0, g_j(\mathbf{x}_n)\} + \sum_{j=1}^{H} |h_j(\mathbf{x}_n)| & \text{else} \end{cases} \tag{12}$$

The particle's best ever position, $\mathbf{p}_n$, and the swarm's best ever position, $\mathbf{s}$, need to be updated, which is done by comparing the current position with the best positions by the following rules:

1. if current and best positions are feasible, the one with best fitness wins
2. if either the current or best positions are feasible and the other infeasible, the feasible position wins
3. if current and best positions are infeasible, the one with the minimum constraint violation wins

For the N_f feasible particles only, the mass of the i-th feasible particle is calculated based on a particle's feasible fitness:

$$m_i(t) = \frac{f(\mathbf{x}_i(t)) - \max f(\mathbf{x}(t))}{\min f(\mathbf{x}(t)) - \max f(\mathbf{x}(t))} \tag{13}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N_f} m_j(t)} \tag{14}$$

where the best and worst fitnesses are from the feasible particles only. The force acting on particle i from particle j, where both particles are feasible is:

$$F_{i,j}^d(t) = G(t) \left(\frac{M_i(t) M_j(t)}{R_{i,j}(t) + \epsilon} \right) (x_j^d(t) - x_i^d(t)) \tag{15}$$

$$G(t) = G_0 \exp(-\alpha t / T) \tag{16}$$

The total force acting on the i-th feasible particle is:

$$F_i^d(t) = \sum_{j=1, j \neq i}^{\min\{N_f, Kbest\}} \text{rand}_j F_{i,j}^d(t) \tag{17}$$

where a random value (uniformly distributed between 0 and 1) is used to allow variability and full design space exploration. The acceleration of the feasible particles uses the cognitive and social memory parameters of particle swarm combined with the global transfer of data through the force mechanism of gravitational search to facilitate faster convergence:

$$a_i^d(t)_{gsa} = F_i^d(t) / M_i(t) \tag{18}$$

$$a_i^d(t)_{pso} = c_1 r_{1_i} (p_i^d - x_i^d(t)) + c_2 r_{2_i} (s^d - x_i^d(t)) \tag{19}$$

$$a_i^d(t) = (a_i^d(t)_{gsa} + a_i^d(t)_{pso}) / 2 \tag{20}$$

where $\mathbf{p}_i$ and $\mathbf{s}$ are the particle's and swarm's best positions ever, which are always feasible in this part of the optimization process. The cognitive, c_1, and social, c_2, parameters do add additional parameters to the problem but it is expected that these have the same value as the cognitive and social parameters is the infeasible search (equation 21).

The acceleration of the j-th infeasible particles is done by a pure particle swarm approach:

$$a_j^d(t) = c_1 r_{1_j}(p_j^d - x_j^d(t)) + c_2 r_{2_j}(s^d - x_j^d(t)) \tag{21}$$

The updating procedure for all of the particles is:

$$v_n^d(t+1) = \text{rand}_n v_n^d(t) + a_n^d(t) \tag{22}$$

$$x_n^d(t+1) = x_n^d(t) + v_n^d(t+1) \tag{23}$$

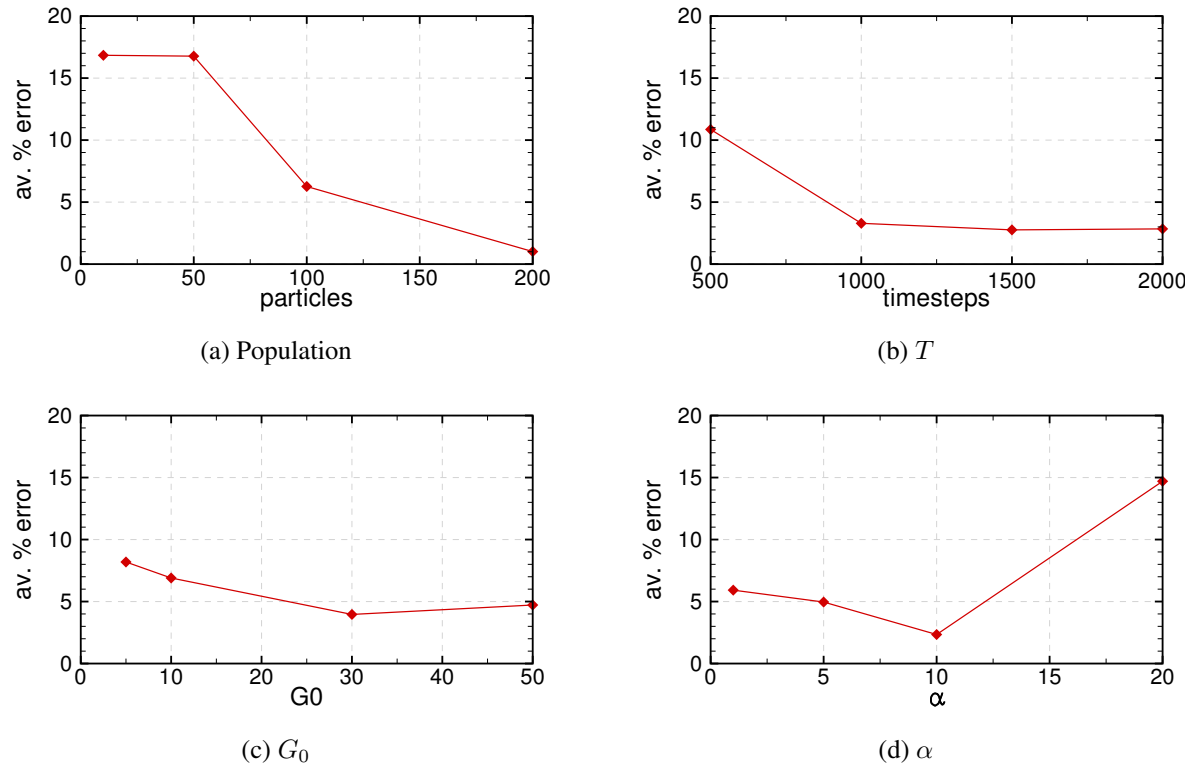
If a particle exceeds the boundary of the search space then this is not an infeasible particle but is a particle without a solution so is reinitialised in its last position with a zero velocity.

The 3S-GSA algorithm is independent of objective function evaluation, so for the position of any given agent, that agent need only know a value for the objective function. In the aerodynamic shape optimization process implemented here the objective function evaluation requires one flow solution to obtain the aerodynamic forces which are then used to evaluate the objective and constraints. The nature of agent-based search systems means that the objective of particles is independent, so for any given particle it only requires a flow solution from the perturbed mesh that relates to its position, and this makes the algorithm ideal for parallelisation. The implementation of the 3S-GSA algorithm parallelises the objective function evaluations of the particles, which is done in the MPI environment. The algorithm is independent of objective function so the agents call a wrapper which perturbs the surface and CFD mesh, followed by a flow solver to obtain the forces which form the objective function and constraints. Once the agents have the objective function and constraint values, the master processor carries out the position update by the 3S-GSA algorithm.

4.2. Parameter Selection

To investigate, and ensure correct parameters for the optimization are set, the 3S-GSA optimization algorithm has been run on a suite of analytical test functions that are commonly used to benchmark optimization algorithms. The suite, as outlined by Michalewicz and Schoenauer [54] contains 11 analytical test cases that are all minimization problems and contain various numbers

of linear and non-linear inequality and equality constraints, various sizes of feasible search space, and various types of objective function. The solutions of the problems are known so it is possible to analyse the performance of the algorithm exactly. Variations in particle number, maximum number of optimizer iterations, initial gravitational constant and decay constant are shown in figure 6. The settings chosen reflect the results shown. It should be noted that a smaller particle number (96) was chosen as this reflected the best trade-off between computational cost and performance. The optimizer settings used are summarised in table 1.



(a) Population (b) T

(c) G_0 (d) α

Figure 6: Effect of changing constants in 3S-GSA.

5. Flow Solver

The flow-solver used is a structured multiblock, finite-volume, unsteady, cell-centred scheme solving the compressible Euler or Reynolds-Averaged Navier-Stokes (RANS) equations in cartesian and rotating coordinate system. The convective terms are evaluated using

Table 1: 3S-GSA parameters

Parameter	Value
Population	96
T	1500
G_0	30
α	10
c_1	2.0
c_2	2.0

third-order upwind spatial approximation with the flux vector splitting of van Leer [55]. Diffusive terms are evaluated using second-order central differences, and turbulent viscosity is modelled by the Spalart-Allmaras one-equation model [56, 57]. Multi-stage Runge-Kutta with local timestepping is used for time integration, and convergence acceleration is achieved through V-cycle multigrid [58].

For inviscid flows, single block O-meshes were generated using a conformal mapping approach. Figure 7 show views of the 257×97 point meshes, which extend to 100 chords at farfield, of the NACA0012 and RAE2822 aerofoils respectively. This density of mesh is referred to as the L3 mesh and is what the optimization is run on. For the mesh convergence studies presented below, the L1 mesh represents a refinement on the base L3 mesh by two levels. All surface cells have an aspect ratio of one.

For the viscous case considered (see table 2), a three-block C-mesh was generated. Figure 8 shows the blocking structure and two views of the mesh which has 385 points around the aerofoil, 97 points along the wake line and 129 points into the farfield. The initial cell height was chosen to ensure $y+ \approx 1$ for the case considered.

6. Aerofoil Optimization

To test the performance of the mathematically extracted design variables, a number of two-dimensional aerofoil inviscid and viscous drag minimization optimizations are performed. The

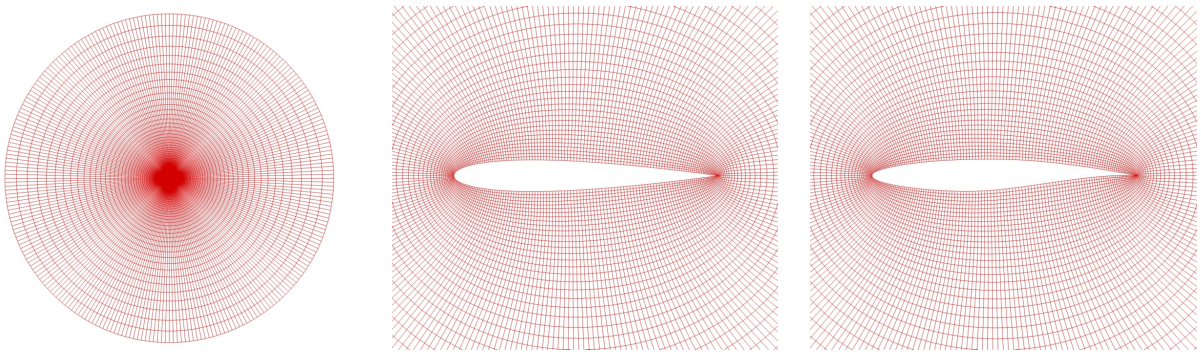


Figure 7: 257×97 NACA0012 and RAE2822 O-meshes

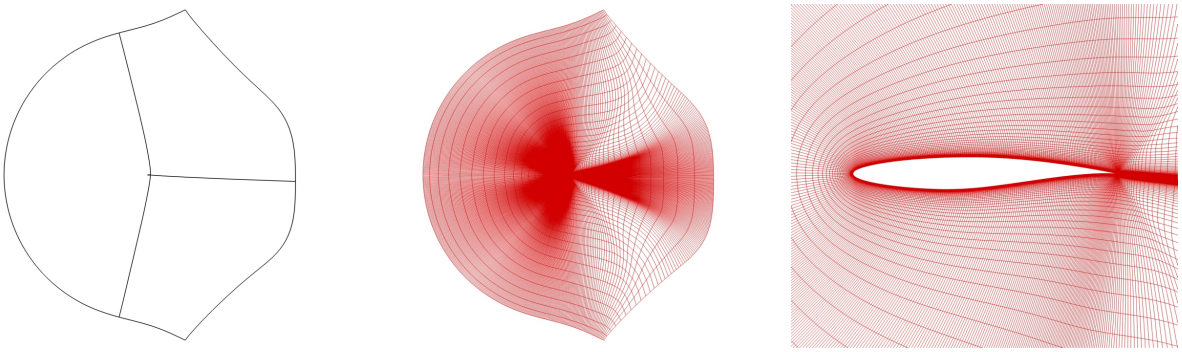


Figure 8: Three-block RAE2822 viscous C-mesh

cases tested and the optimization results are detailed below.

6.1. Cases Tested

The aerofoil optimization cases considered are given in table 2. These five cases each have differing shock strength with different initial aerofoils or a different design problem. Case 1 had been used previously to consider the performance of the domain element method using design variables that have been designed using user intuition [11]. Case 2 is introduced to investigate the effect of a different starting aerofoil to case 1, but for the same flow conditions. Case 3 provides different flow conditions to case 1 for the same given aerofoil. Case 4 is a symmetric zero lift drag minimization case and represents a difficult optimization case. **Finally, case 5 is a viscous optimization based on that performed in the investigations of Chernukhin and Zingg [15]. The angle of attack was chosen to ensure a consistent lift coefficient, which was** $C_l = 0.69$**.**

0.1% deviations were allowed to be made to the constraints during the optimization to allow full flexibility.

The first four cases are inviscid drag minimization cases. As such, the optimal solution is any solution that eliminates wave drag. For these types of problem, there is no requirement for the final solution to have non-oscillatory pressure distributions, and as long as the shock is eliminated then the optimizer can no longer improve the aerofoil performance. Therefore it is likely that the final pressure distributions could feasibly be oscillatory. To design against this, each of the four inviscid cases was run as the pure drag minimization case, but also as a drag minimization with a penalty added to increase the objective function in the event of oscillations in the pressure distributions. This penalty is the total absolute curvature of the pressure distribution around the aerofoil. Total absolute curvature gives a measure of the variations of the pressure coefficient and is therefore a suitable choice for the penalty formulation. The two objective functions (denoted by J) are given below.

$$J = C_d \qquad , \qquad J = C_d + \kappa = C_d + \oint |C_p''| ds$$

Oscillations in the surface or in the pressure coefficient for the viscous optimization case are not an issue due to the added viscous drag or flow separation that arises. As such, the viscous case (case 5) is run as a pure drag minimization case with the appropriate constraints.

6.2. Design Variables

The optimization design variables are $\boldsymbol{\beta}$ in equation 5. These are the scalings of the modal deformations. Different numbers of aerofoil deformation modes were tested to assess the impact of dimensionality on the optimization results, and these are 6, 8 and 10. Larger numbers of design variables are generally avoided for global search algorithms. The use of the modal parameters as design variables therefore allow the use of global search algorithms which can otherwise be prohibitive for many shape deformation techniques. The specific modal parameters selected are taken based on those with the highest energy from the $\boldsymbol{\Sigma}$ matrix, hence if six modes are used then these are the six modes with the highest energy. It should therefore be noted that using different

Table 2: Test cases considered

Case	Aerofoil	M_∞	α	Re	Objective	Constraints		
						Lift	Moment	Area
1	NACA0012	0.65	5.0	-	C_d $C_d + \kappa$	$C_l \geq C_l(0)$	$\lvert C_m \rvert \leq \lvert C_m(0) \rvert$	$A \geq A(0)$
2	RAE2822	0.65	5.0	-	C_d $C_d + \kappa$	$C_l \geq C_l(0)$	$\lvert C_m \rvert \leq \lvert C_m(0) \rvert$	$A \geq A(0)$
3	NACA0012	0.70	3.0	-	C_d $C_d + \kappa$	$C_l \geq C_l(0)$	$\lvert C_m \rvert \leq \lvert C_m(0) \rvert$	$A \geq A(0)$
4	NACA0012	0.85	0.0	-	C_d $C_d + \kappa$	$C_l = 0.0$	$\lvert C_m \rvert = 0.0$	$A \geq A(0)$
5	RAE2822	0.729	2.05	7.0×10^6	C_d	$C_l \geq C_l(0)$	$\lvert C_m \rvert \leq \lvert C_m(0) \rvert$	$A \geq A(0)$

numbers of design variables is a true dimensionality study as larger numbers of design variables contain, exactly, the design space of smaller numbers of design variables.

To further compare the effect of using the modal parameters, they are compared to deforming the control points individually. Using a set of 24 surface control points therefore means that there are 24 individual design variables. For case 4, which is a symmetric deformation cases, the control points deform symmetrically resulting in 11 design variables (the control points at the leading and trailing edge do not move in this case). For all cases, except case 4, a global pitch parameter was added to allow balancing of the lift.

6.3. Results

The four inviscid optimization cases outlined in table 2 have been run with both objective functions with the 3S-GSA algorithm using all of the design variables (modes and individual perturbations), while the viscous case has been run for pure drag minimization. Results for the inviscid cases are shown for each case individually in tables 3 to 6, where the L3 mesh is the one used for the optimization, and the L1 optimization is a refined mesh. **The results of the viscous case are given in table 7.** Surface shapes and pressure coefficients on the surface are shown in figures 9 to 13 for the lowest drag optimization in each of the five cases.

Table 3: Case 1 optimization results (C_d in counts)

	$J = C_d$					$J = C_d + \kappa$					
		L3 mesh		L1 mesh			L3 mesh		L1 mesh		
Variables	C_l	C_d	ΔC_d	C_d	ΔC_d	C_l	C_d	ΔC_d	C_d	ΔC_d	ΔJ
Initial	0.8231	201.4	**-**	188.5	**-**	0.8231	201.4	**-**	188.5	**-**	-
6 modes	0.8231	24.4	**-87.6%**	16.4	**-91.3%**	0.8231	28.4	**-85.9%**	16.6	**-91.2%**	-75.7%
8 modes	0.8231	23.7	**-88.0%**	17.1	**-91.0%**	0.8231	27.9	**-86.1%**	19.9	**-89.4%**	-77.7%
10 modes	0.8231	20.6	**-89.6%**	16.0	**-91.5%**	0.8231	24.1	**-88.0%**	16.9	**-91.0%**	-84.8%
24 Individual	0.8231	22.8	**-88.5%**	16.5	**-91.3%**	N/A	N/A	**N/A**	N/A	**N/A**	N/A

Table 4: Case 2 optimization results (C_d in counts)

	$J = C_d$					$J = C_d + \kappa$					
		L3 mesh		L1 mesh			L3 mesh		L1 mesh		
Variables	C_l	C_d	ΔC_d	C_d	ΔC_d	C_l	C_d	ΔC_d	C_d	ΔC_d	ΔJ
Initial	1.208	241.4	**-**	233.7	**-**	1.208	241.4	**-**	233.7	**-**	-
6 modes	1.208	42.0	**-82.6%**	35.9	**-84.7%**	1.208	44.2	**-81.7%**	25.3	**-89.2%**	-71.3%
8 modes	1.208	41.3	**-82.9%**	35.0	**-85.0%**	1.208	42.5	**-82.4%**	24.9	**-89.3%**	-71.5%
10 modes	1.208	40.2	**-83.3%**	34.7	**-85.1%**	1.208	42.1	**-82.6%**	24.6	**-89.5%**	-74.9%
24 Individual	1.208	41.8	**-82.7%**	35.4	**-84.9%**	N/A	N/A	**N/A**	N/A	**N/A**	N/A

Table 5: Case 3 optimization results (C_d in counts)

	$J = C_d$					$J = C_d + \kappa$					
		L3 mesh		L1 mesh			L3 mesh		L1 mesh		
Variables	C_l	C_d	ΔC_d	C_d	ΔC_d	C_l	C_d	ΔC_d	C_d	ΔC_d	ΔJ
Initial	0.5601	92.8	**-**	85.8	**-**	0.5601	92.8	**-**	85.8	**-**	-
6 modes	0.5601	17.3	**-80.3%**	7.89	**-90.8%**	0.5601	21.9	**-76.4%**	8.29	**-90.3%**	-55.2%
8 modes	0.5601	16.1	**-81.6%**	9.11	**-89.4%**	0.5601	18.7	**-79.8%**	8.78	**-89.8%**	-54.3%
10 modes	0.5601	12.8	**-85.4%**	7.73	**-91.0%**	0.5601	20.4	**-78.0%**	7.81	**-90.9%**	-68.1%
24 Individual	0.5601	18.5	**-80.1%**	9.92	**-88.4%**	N/A	N/A	**N/A**	N/A	**N/A**	N/A

Table 6: Case 4 optimization results (C_d in counts)

	$J = C_d$					$J = C_d + \kappa$					
		L3 mesh		L1 mesh			L3 mesh		L1 mesh		
Variables	C_l	C_d	ΔC_d	C_d	ΔC_d	C_l	C_d	ΔC_d	C_d	ΔC_d	ΔJ
Initial	0.0	467.54	**-**	465.14	**-**	0.0	467.54	**-**	465.14	**-**	-
6 modes	0.0	7.75	**-98.3%**	2.18	**-99.5%**	0.0	7.81	**-98.3%**	1.65	**-99.6%**	-92.0%
8 modes	0.0	6.79	**-98.5%**	1.05	**-99.8%**	0.0	8.65	**-98.2%**	1.96	**-99.6%**	-92.1%
10 modes	0.0	6.75	**-98.6%**	1.11	**-99.8%**	0.0	8.55	**-98.2%**	2.05	**-99.6%**	-93.7%
24 Individual	0.0	8.60	**-98.2%**	1.41	**-99.7%**	N/A	N/A	**N/A**	N/A	**N/A**	N/A

Table 7: Case 5 optimization results (C_d in counts)

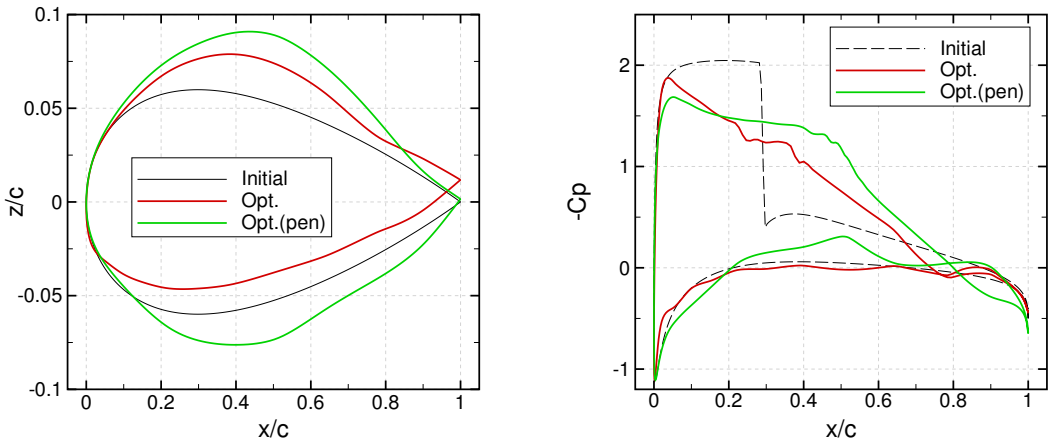
Variables	C_l	$C_{d_{pres}}$	$C_{d_{visc}}$	C_d	ΔC_d
Initial	0.69	57.4	53.5	110.9	**-**
6 modes	0.69	44.9	54.0	98.9	**-10.8%**
8 modes	0.69	45.5	53.4	98.9	**-10.8%**
10 modes	0.69	44.0	54.0	98.0	**-11.6%**
24 Individual	0.69	46.1	53.8	99.9	**-9.9%**

The final optimized solutions shown in tables 3 to 7 demonstrate the high performance of the mathematically derived modal parameters as design variables. There is convergence of the modes for increasing numbers of design variables, such that a better optimization result is always obtained using higher numbers of modal parameters for the mesh used for optimization (L3 mesh). It is interesting to note that when refining the mesh after performing the optimization, that this is not necessarily the case, however, the performance is still better than the alternative set of design variables tested. **It should, however, be noted that once the shock is eliminated from the inviscid cases, the remaining drag is numerical so further reductions in the headline drag number arise due to the optimizer using the modal deformations to reduce the causes of the numerical drag. For case 2, where (as explained below) a shock is still present, the reduction in the headline drag value results from the higher number of modes having sufficient fidelity of surface deformation to be able to reduce the strength of the shock.** For all of the cases tested, the lowest drag result obtained was always an optimization using the modal parameters as design variables, while all optimizations produce results where the lift constraint is active, as expected.

The optimized surface shapes and pressure distributions with lowest drag for each case are given in figures 9 to 13. All of the solutions appear shock free, except for case 2 where a weak shock remains. This is a highly loaded aerofoil (C_l =1.208) so a shock free-solution may not be possible.

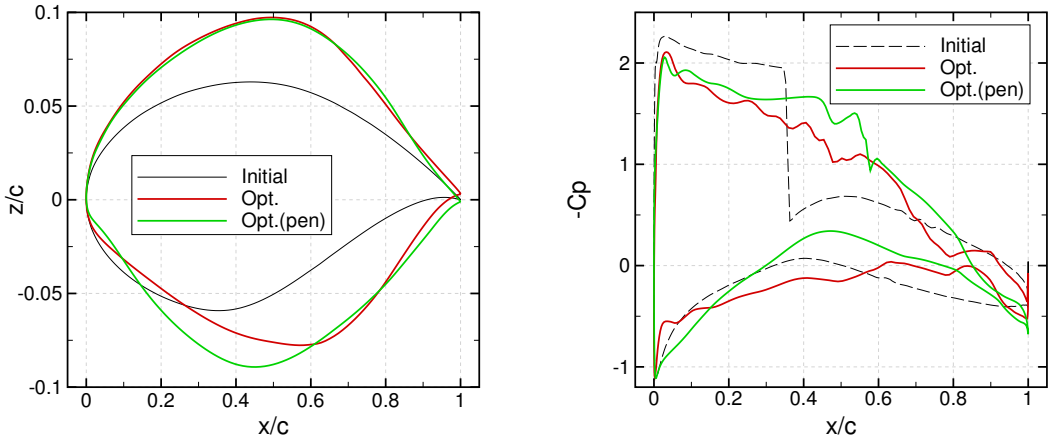
When the optimization is performed with the curvature penalty added, the pressure distributions become slightly less oscillatory, primarily on the lower surface of the aerofoils. **Despite this, for the inviscid optimizations (cases 1 to 4) oscillations in the pressure distribution still remain. It is worth noting, however, that these are an artefact of performing inviscid optimization. In comparison to the viscous optimizations (an example pressure distribution is shown in figure 13), performing the raw drag minimization problem has resulted in an entirely smooth pressure distribution. All optimizations were performed using the same modal deformations hence, when viscous effects are considered, the expected smoother pressure distribution results.**

While the pressure distributions for the inviscid optimizations with the penalty are better be-



(a) Surface (b) C_p

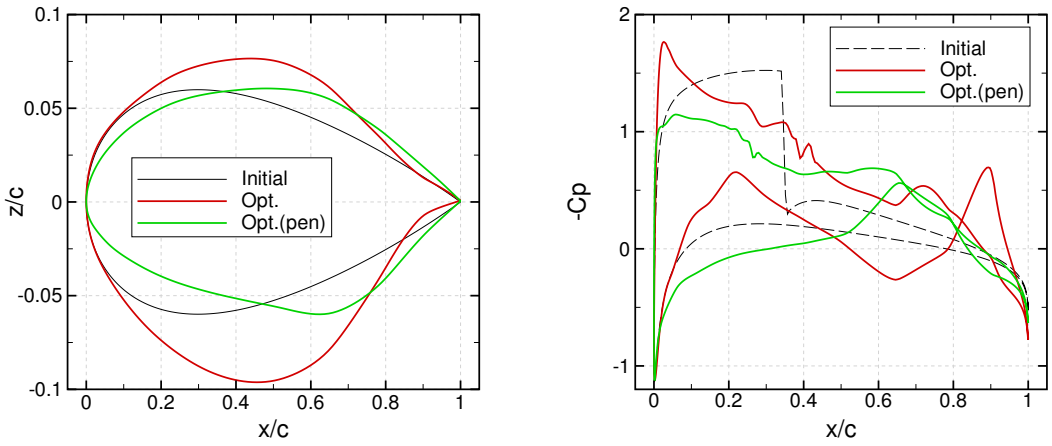
Figure 9: Optimum surface shapes and pressure distributions for case 1



(a) Surface (b) C_p

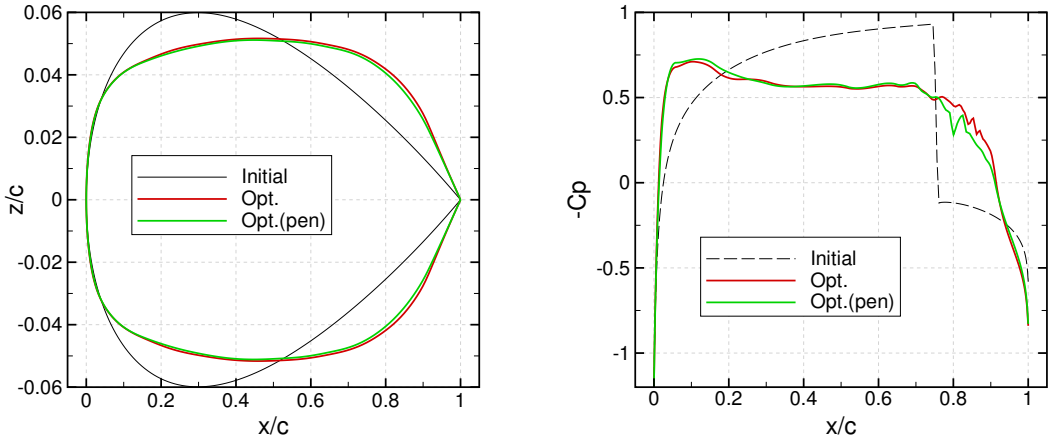
Figure 10: Optimum surface shapes and pressure distributions for case 2

haved, there is a trade-off between smoothness and drag. In almost all cases, the final drag values obtained when performing the optimizations with the smoothness penalty results in slightly higher drag values. **Furthermore, when considering the objective with the pressure curvature penalty the convergence of the optimizer is severely degraded, possibly indicating the presence of multimodality for this problem.** Combining the curvature penalty with individual perturbations can cause large changes in the value of the curvature penalty in a local region.



(a) Surface (b) C_p

Figure 11: Optimum surface shapes and pressure distributions for case 3

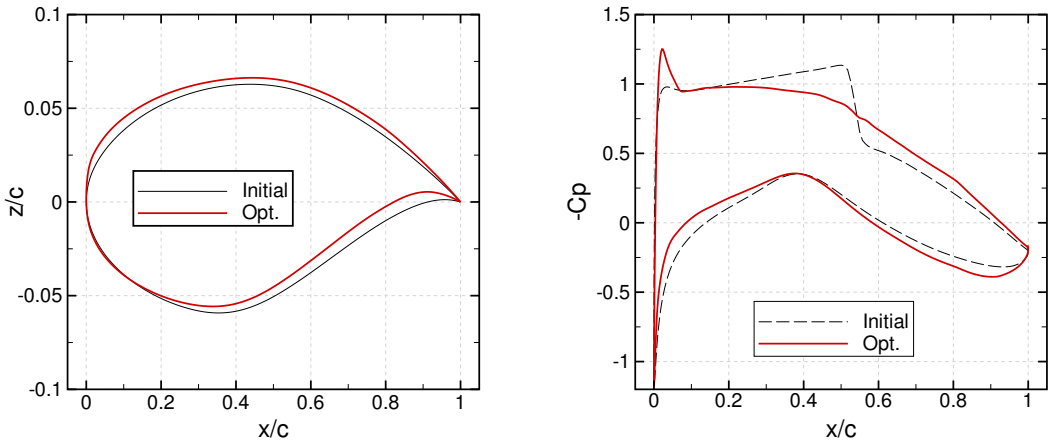


(a) Surface (b) C_p

Figure 12: Optimum surface shapes and pressure distributions for case 4

This can therefore cause a large value for the global curvature penalty and therefore cause severe oscillations in the objective function for only small changes in one surface perturbation. Using individual control point deformations is therefore not conducive with the curvature penalty. Using the modal parameters alleviates this as the changes are global so cause a smoothing effect on the curvature penalty.

As noted in section 3.1, the aerofoil library from which the modes are extracted is based

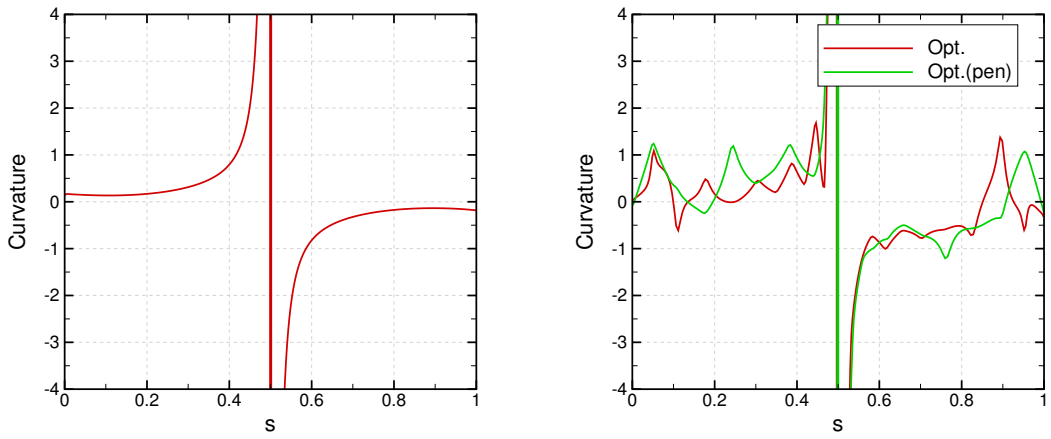


(a) Surface

(b) C_p

Figure 13: Optimum surface shapes and pressure distributions for case 5

on the raw data from an online database, which is of mixed quality, however, for consistency, the aerofoil surfaces from which the SVD modes are derived are all driven exactly through the raw data. By applying the SVD, the resulting lower frequency modes have some of these issues filtered out. As shown in figure 14, which shows the curvature of the initial and optimized aerofoils for case 1, while the curvature is oscillatory which leads to the oscillatory pressure distributions, it is reasonably well behaved. Adding the pressure coefficient curvature penalty has had the effect of reducing the pressure coefficient oscillations but also the oscillations in the curvature as expected. Further curvature-based smoothing can also be performed by, for example, the method of Li and Krist [59]. Alternatively, the initial aerofoil library can also be smoothed, as was done by Masters *et al.* [21].



(a) Initial aerofoil

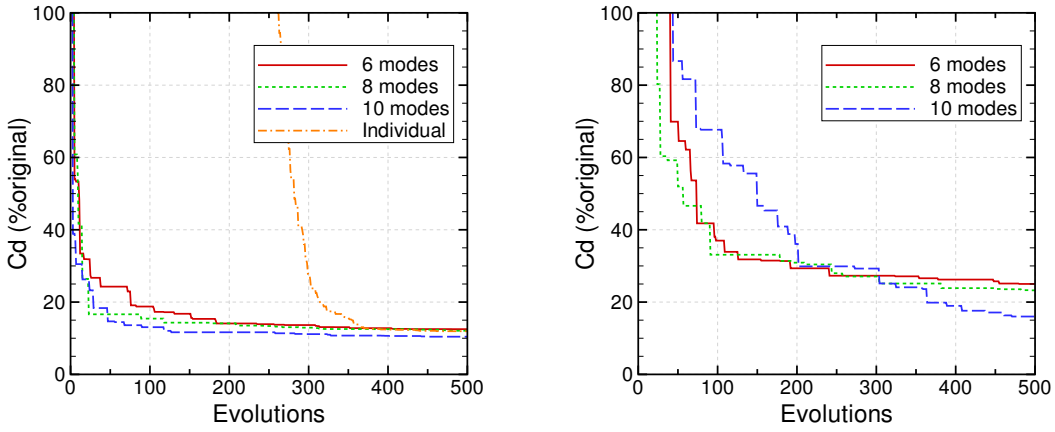
(b) Optimized aerofoils

Figure 14: Surface curvatures along normalised periphery for case 1

A typical optimizer convergence is shown in figure 15 (this is for case 1). This demonstrates that in the first few hundred iterations, exploration of the design space was the driving force, and once a global optimum region is located, exploitation of that optimum occurs. From around iteration 500 to the maximum number of iterations, the objective function changes are very small and often do not change. **The convergence of the curvature penalty objective is considerably slower than the pure drag objective which could be indicative a more oscillatory design space when the penalty function is added.** Furthermore, the convergence of the modal parameters is faster than the individual control point perturbations. This is due to the optimization using fewer design variables and having a simpler design space, which comes about due to having orthogonal design variables from the singular value decomposition. This further emphasises that the use of orthogonal design variables can be advantageous for aerodynamic shape optimization. Fast convergence to small drag values is observed, indicating the high performance of the global constrained optimization algorithm developed for these optimizations.

This behaviour is also demonstrated in figure 16 which shows the positions of the particles in two of the 11 design variables of the 10 mode case 1 optimization; these are weightings with the highest values. In all cases, the design space is normalised to a hypercube $[-1, 1]^n$, where n is the number of parameters. The spread of particles over the design space is large up around 200

evolutions, but from then onwards the particles converge on an optimal solution. From around iteration 500 to the maximum number of iterations, the objective function changes are very small and often do not change. The convergence of the modal parameters is faster than the individual control point perturbations and is due to the optimization using fewer design variables and having a simpler design space. Fast convergence to small drag values is observed, indicating the high performance of the global constrained optimization algorithm developed for these optimizations.

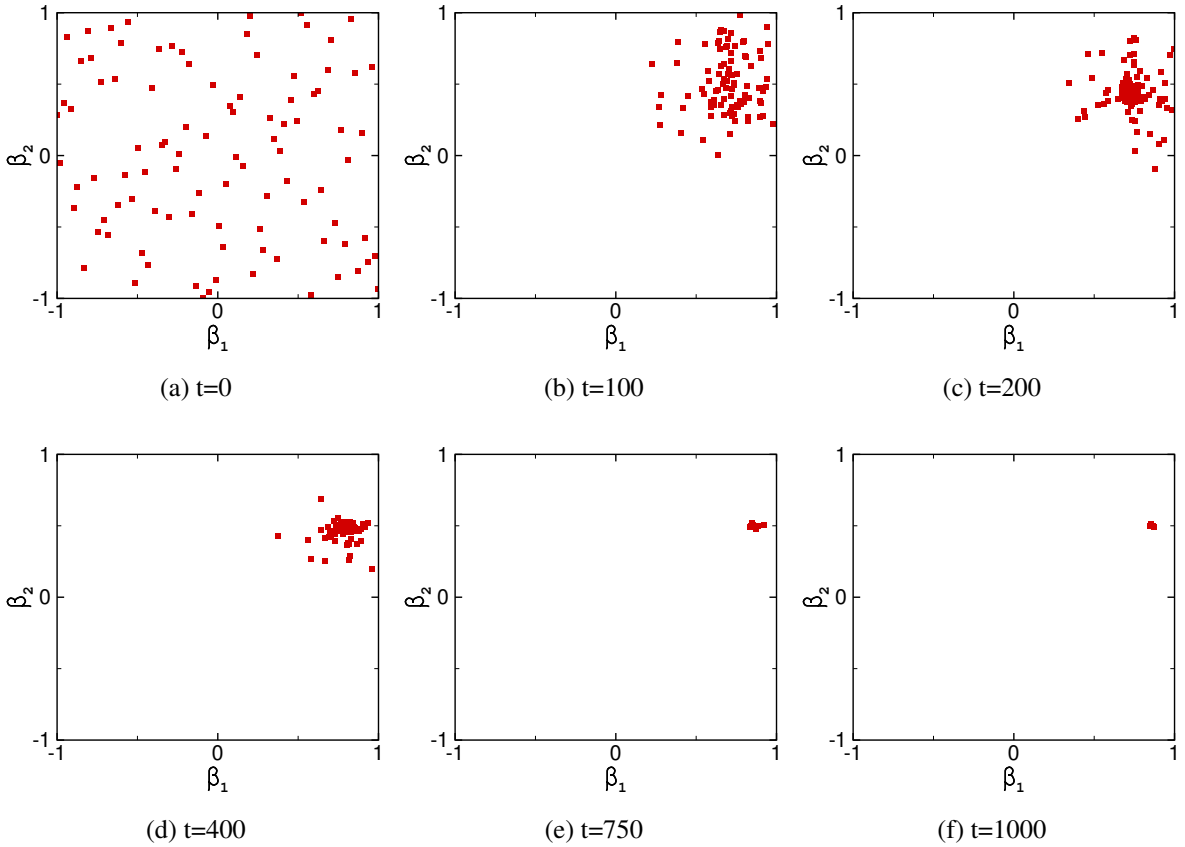


(a) Without penalty

(b) With penalty

Figure 15: Optimizer convergence of case 1 for first 500 iterations

A mesh convergence study of the 10 mode optimized solutions for the inviscid cases is presented in figure 17. Each refinement represents a doubling of the base grid in each direction from the optimized solution. This ensures continuity between the optimized solution and the mesh convergence study. Figure 17 shows that further resolution of the flowfield results in lower drag values. **The results for case 4 demonstrate that refining towards a continuous mesh produces almost zero drag, indicating that the result is a truly shock-free result. However, it is not necessarily the case that a refined mesh from an initial, coarser, shock-free solution will be shock-free, as demonstrated by Jameson and Vassberg [60]. In the other shock-free cases (1 and 3), while shock-free solutions have been obtained for the mesh density used for optimization, refined meshes exhibit slightly shocked solutions, hence a small amount of drag is present. This does not detract from the performance of the optimization methodology considered**

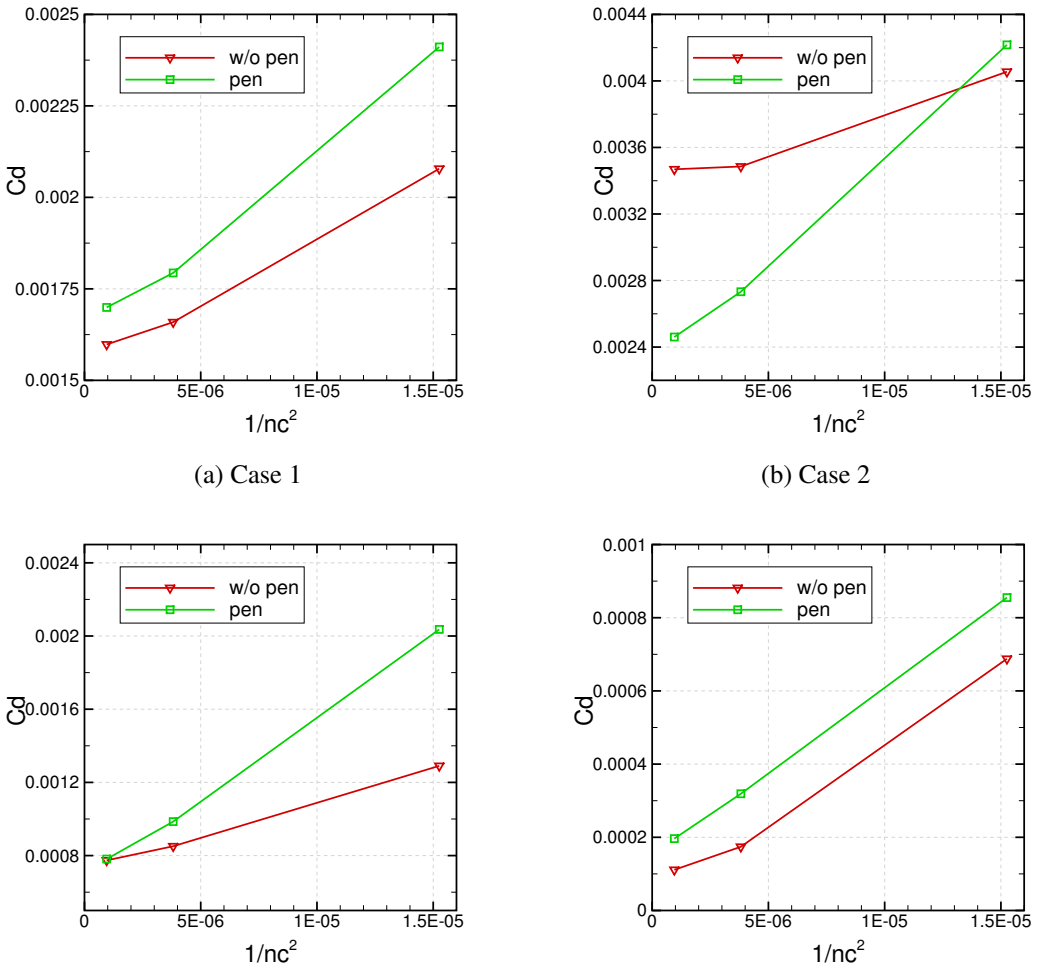


(a) t=0 (b) t=100 (c) t=200

(d) t=400 (e) t=750 (f) t=1000

Figure 16: Example of particle convergence in two-dimensions for 10 modes case 1

here (since shock-free solutions were found on the mesh density used for optimization), but rather raises an important issue regarding the construction of the aerodynamic optimization problem.



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 17: Mesh convergence (grid used for optimization is right-most point)

7. Concluding Remarks

Optimization of aerofoils using efficient orthogonal design variables has been considered using a global search algorithm. Aerofoil design variables have been extracted by a singular value decomposition (SVD) approach where a training library of aerofoils is analysed and decomposed to obtain an efficient and reduced set of design variables. Using the SVD allows the optimal representation of the original library in a compact set of deformation modes that represent typical aerofoil design parameters such as thickness and camber.

The performance of the modal design variables have been demonstrated on a set of challenging

inviscid **and viscous** aerofoil optimization test cases. To reduce, as much as possible, the question of multimodality in these problems, a global search algorithm with an effective constraint handling method has been developed and is used to perform the optimizations. An issue with global search algorithms is often the so-called 'curse of dimensionality' where the performance of these algorithms degrades with increasing numbers of design variables, which can be a barrier to their use for aerodynamic shape optimization problems. The use of the efficient decomposition-derived design variables, however, reduces this problem substantially by allowing few design variables to be used (10 or fewer) without significant degrading of optimization capability; results have shown that shock free solutions are obtainable in most test cases with 10 or fewer design variables, which is a significant result. A pressure coefficient curvature penalty has also be considered for inviscid optimizations to force a less oscillatory pressure distribution. **Results indicate that this is a method that can produce a less oscillatory pressure distribution, which is otherwise common in aerofoils optimized in inviscid flow, however, the convergence of the optimizer is slower so could be indicative of a multimodal design space.** For the viscous optimization using the same modal deformations, an oscillatory pressure distribution is avoided while still producing a shock-free solution. Convergence of the optimizations towards lower drag values as the number of modal parameters is increased has been demonstrated, and is indicative of the use of orthogonal parameters; the design space of fewer modes is entirely contained within the design space of a greater number of modes.

Extension of using orthogonal modal design parameters in three-dimensional optimization is to be considered in future work. Commonly, using aerofoil design parameters in wing optimization (for example) will implement sectional-based optimization of the wing sections combined with spanwise blending to ensure a smooth surface, and this can be combined with planform design parameters too. However, to exploit the advantages of orthogonality that comes with using the derived modal parameters, care must be taken in this regard.

References

[1] R. M. Hicks, P. A. Henne, Wing design by numerical optimization, Journal of Aircraft 15 (7) (1978) 407–412. doi:10.2514/6.1977-1247.

[2] W. Li, L. W. Huyse, S. Padula, Robust aerofoil optimization to achieve consistent drag reduction over a Mach range, Structural and Multidisciplinary Optimization 24 (1) (2002) 38–50. doi:10.1007/s00158-002-0212-4.

[3] N. Qin, A. Vavalle, A. Le Moigne, M. Laban, K. Hackett, P. Weinerfelt, Aerodynamic considerations of blended wing body aircraft, Progress in Aerospace Sciences 40 (6) (2004) 321–343. doi:10.1016/j.paerosci.2004.08.001.

[4] W. S. Wong, A. Le Moigne, N. Qin, Parallel adjoint-based optimisation of a blended wing body aircraft with shock control bumps, The Aeronautical Journal 111 (1117) (2006) 165–174.

[5] Z. Lyu, G. K. W. Kenway, J. R. R. A. Martins, Aerodynamic shape optimization investigations of the common research model wing benchmark, AIAA Journal 53 (4) (2015) 968–985. doi:10.2514/1.J053318.

[6] A. Jameson, J. C. Vassberg, S. Shankaran, Aerodynamic-structural design studies of low-sweep transonic wings, Journal of Aircraft 47 (2) (2010) 505–514. doi:10.2514/1.42775.

[7] A. Le Pape, P. Beaumier, Numerical optimization of helicopter rotor aerodynamic performance in hover, Aerospace Science and Technology 9 (3) (2005) 191–201. doi:10.1016/j.ast.2004.09.004.

[8] C. Tatossian, S. K. Nadarajah, P. Castonguay, Aerodynamic shape optimization of hovering rotor blades using a Non-Linear Frequency Domain approach, Computers and Fluids 51 (1) (2011) 1–15. doi:10.1016/j.compfluid.2011.06.014.

[9] E. J. Nielsen, E. M. Lee-Rausch, W. T. Jones, Adjoint based design of rotors in a noninertial frame, Journal of Aircraft 47 (2) (2010) 638–646. doi:10.2514/1.46044.

[10] S. Choi, K. H. Lee, M. Potsdam, J. J. Alonso, Helicopter rotor design using a time-spectral and adjoint based method, Journal of Aircraft 51 (2) (2014) 412–423. doi:10.2514/1.C031975.

[11] A. M. Morris, C. B. Allen, T. C. S. Rendall, CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation, International Journal for Numerical Methods in Fluids 58 (8) (2008) 827–860. doi:10.1002/fld.1769.

[12] C. B. Allen, T. C. S. Rendall, Computational-fluid-dynamics-based optimisation of hovering rotors using radial basis functions for shape parameterisation and mesh deformation, Optimization and Engineering 14 (2013) 97–118. doi:10.1007/s11081-011-9179-6.

[13] M. Imiela, High-fidelity optimization framework for helicopter rotors, Aerospace Science and Technology 23 (2012) 2–16. doi:10.1016/j.ast.2011.12.011.

[14] J. E. Hicken, D. W. Zingg, Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement, AIAA Journal 48 (2) (2010) 400–413. doi:10.2514/1.44033.

[15] O. Chernukhin, D. W. Zingg, Multimodality and global optimization in aerodynamic design, AIAA Journal 51 (6) (2013) 1342–1354. doi:10.2514/1.J051835.

[16] A. Jahangirian, A. Shahrokhi, Aerodynamic shape optimization using efficient evolutionary algorithms and

unstructured CFD solver, Computers and Fluids 46 (2011) 270–276. doi:10.1016/j.compfluid.2011.02.010.

[17] B. Epstein, S. Peigin, Optimization of 3D wings based on navier-stokes solutions and genetic algorithms, International Journal of Computational Fluid Dynamics 20 (2) (2006) 75–92. doi:10.1080/10618560600761601.

[18] L. Blasi, G. Del Core, Particle swarm approach for finding optimum aircraft configuration, Journal of Aircraft 44 (2) (2007) 679–682. doi:10.2514/1.24399.

[19] P. Castonguay, S. K. Nadarajah, Effect of shape parameterization on aerodynamic shape optimization, in: 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2007, AIAA Paper 2007-59. doi:10.2514/6.2007-59.

[20] D. J. Poole, C. B. Allen, T. C. S. Rendall, Metric-based mathematical derivation of efficient airfoil design variables, AIAA Journal 53 (5) (2015) 1349–1361. doi:10.2514/1.J053427.

[21] D. A. Masters, N. J. Taylor, T. C. S. Rendall, C. B. Allen, D. J. Poole, A geometric comparison of aerofoil shape parameterisation methods, in: 54th AIAA Aerospace Sciences Meeting, San Diego, California, 2016, AIAA Paper 2016-0558. doi:10.2514/6.2016-0558.

[22] J. A. Samareh, Status and future of geometric modelling and grid generation for design and optimization, Journal of Aircraft 36 (1) (1999) 97–104. doi:10.2514/2.2417.

[23] J. A. Samareh, Survey of shape parameterization techniques for high fidelity multidisciplinary shape optimization, AIAA Journal 39 (5) (2001) 877–884. doi:10.2514/2.1391.

[24] A. Mousavi, P. Castonguay, S. K. Nadarajah, Survey of shape parameterization techniques and its effect on three dimensional aerodynamic shape optimization, in: 18th AIAA Computational Fluid Dynamics Conference, Miami, Florida, 2007, AIAA Paper 2007-3837. doi:10.2514/6.2007-3837.

[25] D. A. Masters, N. J. Taylor, T. C. S. Rendall, C. B. Allen, D. J. Poole, Review of aerofoil parameterisation methods for aerodynamic shape optimisation, in: 53rd AIAA Aerospace Sciences Meeting, Kissimmee, Florida, 2015, AIAA Paper 2015-0761. doi:10.2514/6.2015-0761.

[26] B. M. Kulfan, Universal parametric geometry representation method, Journal of Aircraft 45 (1) (2008) 142–158. doi:10.2514/1.29958.

[27] H. Sobieczky, Parametric airfoils and wings, Notes on Numerical Fluid Mechanics 68 (1998) 71–88.

[28] M. I. G. Bloor, M. J. Wilson, Generating parameterizations of wing geometries using partial differential equations, Computer Methods in Applied Mechanics and Engineering 148 (1997) 125–138. doi:10.1016/S0045-7825(97)00033-9.

[29] V. Braibant, C. Fleury, Shape optimal design using B-splines, Computer Methods in Applied Mechanics and Engineering 44 (3) (1984) 247–267. doi:10.1016/0045-7825(84)90132-4.

[30] F. Zhu, N. Qin, Intuitive class/shape function parameterization for airfoils, AIAA Journal 52 (1) (2014) 17–25.

doi:10.2514/1.J052610.

[31] R. M. Pickett, M. F. Rubinstein, R. B. Nelson, Automated structural synthesis with a reduced number of design coordinates, AIAA Journal 11 (4) (1973) 494–498. doi:10.2514/6.1973-336.

[32] A. Jameson, Aerodynamic design via control theory, Journal of Scientific Computing 3 (3) (1988) 233–260. doi:10.1007/BF01061285.

[33] J. A. Samareh, Novel multidisciplinary shape parameterization approach, Journal of Aircraft 38 (6) (2001) 1015–1024. doi:10.2514/2.2888.

[34] W. Yamazaki, S. Mouton, G. Carrier, Geometry parameterization and computational mesh deformation by physics-based direct manipulation approaches, AIAA Journal 48 (8) (2010) 1817–1832. doi:10.2514/1.J050255.

[35] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (3) (1936) 211–218. doi:10.1007/BF02288367.

[36] I. C. Chang, F. J. Torres, C. Tung, Geometric analysis of wing sections, Tech. rep., NASA Ames Research Centre, Moffett Field, California, NASA Technical Memorandum 110346 (April 1995).

[37] G. M. Robinson, A. J. Keane, Concise orthogonal representation of supercritical airfoils, Journal of Aircraft 38 (3) (2001) 580–583. doi:10.2514/2.2803.

[38] D. J. J. Toal, N. W. Bressloff, A. J. Keane, C. M. E. Holden, Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization, AIAA Journal 48 (5) (2010) 916–928. doi:10.2514/1.41420.

[39] S. S. Ghoman, Z. Wang, P. C. Chen, R. K. Kapania, A POD-based reduced order design scheme for shape optimization of air vehicles, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Co-located Events, Honolulu, Hawaii, 2012, AIAA Paper 2012-1808. doi:10.2514/6.2012-1808.

[40] M. Buhmann, Radial Basis Functions, 1st Edition, Cambridge University Press, 2005.

[41] H. Wendland, Scattered Data Approximation, 1st Edition, Cambridge University Press, 2005.

[42] T. C. S. Rendall, C. B. Allen, Unified fluid-structure interpolation and mesh motion using radial basis functions, International Journal for Numerical Methods in Engineering 74 (10) (2008) 1519–1559. doi:10.1002/nme.2219.

[43] M. Drela, Design and optimization method for multi-element airfoils, in: AIAA/AHS/ASEE Aerospace Design Conference, Irvine, California, 1993, AIAA Paper 1993-969. doi:10.2514/6.1993-969.

[44] D. W. Zingg, M. Nemec, T. H. Pulliam, A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization, European Journal of Computational Mechanics 17 (2008) 103–126. doi:10.3166/remn.17.103-126.

[45] X. Han, D. W. Zingg, An adaptive geometry parameterization for aerodynamic shape optimization, Optimization

and Engineering 15 (1) (2014) 69–91. doi:10.1007/s11081-013-9213-y.

[46] C. A. Mader, J. R. R. A. Martins, Stability-constrained aerodynamic shape optimization of flying wings, Journal of Aircraft 50 (5) (2013) 1431–1449. doi:10.2514/1.C031956.

[47] K. C. Giannakoglou, Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence, Progress in Aerospace Sciences 38 (2002) 43–76. doi:10.1016/S0376-0421(01)00019-7.

[48] H. Namgoong, W. Crossley, A. S. Lyrintzis, Global optimization issues for transonic airfoil design, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, 2002, AIAA Paper 2002-5641. doi:10.2514/6.2002-5641.

[49] M. S. Khurana, H. Winarto, A. K. Sinha, Airfoil optimisation by swarm algorithm with mutation and artificial neural networks, in: 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, Florida, 2010, AIAA Paper 2009-1278. doi:10.2514/6.2009-1278.

[50] H. P. Buckley, B. Y. Zhou, D. W. Zingg, Airfoil optimization using practical aerodynamic design requirements, Journal of Aircraft 47 (5) (2010) 1707–1719. doi:10.2514/1.C000256.

[51] F. van den Bergh, A. P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 225–239. doi:10.1109/TEVC.2004.826069.

[52] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Information Sciences 179 (2009) 2232–2248. doi:10.1016/j.ins.2009.03.004.

[53] D. J. Poole, C. B. Allen, T. C. S. Rendall, A generic framework for handling constraints with agent-based optimization algorithms and application to aerodynamic design, Optimization and EngineeringIn Print.

[54] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1996) 1–32. doi:10.1162/evco.1996.4.1.1.

[55] B. van Leer, Flux-vector splitting for the Euler equations, in: Eighth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, 1982, pp. 507–512. doi:10.1007/3-540-11948-5_66.

[56] P. R. Spalart, S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, Recherche Aérospatiale 1 (1994) 5–21.

[57] S. R. Allmaras, F. T. Johnson, P. R. Spalart, Modifications and clarifications for the implementation of the spalart-allmaras turbulence model, in: Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, 2012, paper ICCFD7-1902.

[58] C. B. Allen, Multigrid convergence of inviscid fixed- and rotary-wing flows, International Journal for Numerical Methods in Fluids 39 (2) (2002) 121–140. doi:10.1002/fld.282.

[59] W. Li, S. Krist, Spline-based airfoil curvature smoothing and its applications, Journal of Aircraft 42 (4) (2005) 1065–1074. doi:10.2514/1.10394.

[60] A. Jameson, J. Vassberg, Further studies of mesh refinement - are shock-free airfoils truly shock free?, in: 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, 2011, AIAA Paper 2011-3983. doi:10.2514/6.2011-3983.